Supplemental Document

Biomedical Optics EXPRESS

Wearable device for remote monitoring of transcutaneous tissue oxygenation: supplement

Juan Pedro Cascales, Emmanuel Roussakis, Lilian Witthauer, Avery Goss, Xiaolei Li, Yenyu Chen, Haley L. Marks, and Conor L. Evans*

Wellman Center for Photomedicine, Massachusetts General Hospital, Harvard Medical School, Charlestown, Massachusetts 02129, USA
*evans.conor@mgh.harvard.edu

Supplement DOI: https://doi.org/10.6084/m9.figshare.13203023

Parent Article DOI: https://doi.org/10.1364/BOE.408850

Wearable device for remote monitoring of transcutaneous tissue oxygenation: supplemental document

LIFETIME CHARACTERIZATION MEASUREMENTS

The lifetime of our oxygen sensing film was measured in room air ($pO_2 = 160mmHg$) and in deoxygenated conditions at room temperature with an Edinburgh Instruments photospectrometer. By fitting a double exponential decay, we obtain that there are two different lifetimes in the room air measurements, with 97% of the phosphors having $\tau = 14.92\mu s$ and the remaining 3% yielding $\tau = 57\mu s$. The 3% is shielded from changes in oxygen, as the deoxygenated measurement gives 100% of emitters having $\tau = 95.73\mu s$.

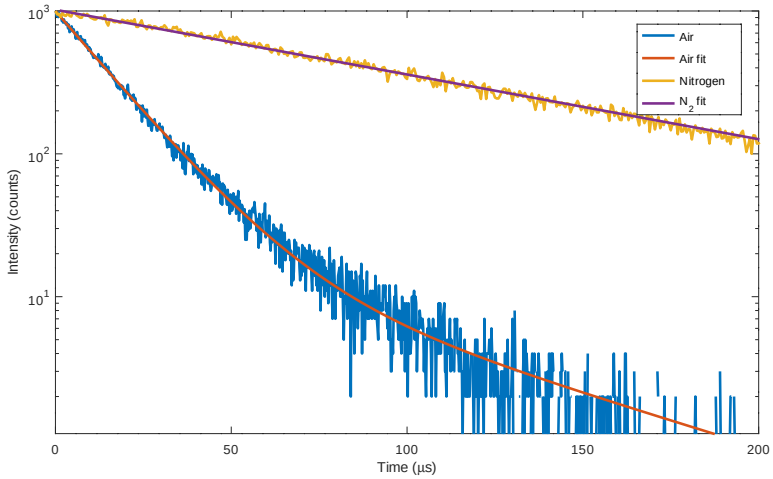

Fig. S1. Exponential decay of the phosphorescence of our oxygen-sensing film in room air and pure nitrogen atmosphere, and a fit to a double exponential decay.

FLEXIBLE OPTICAL FILTERS

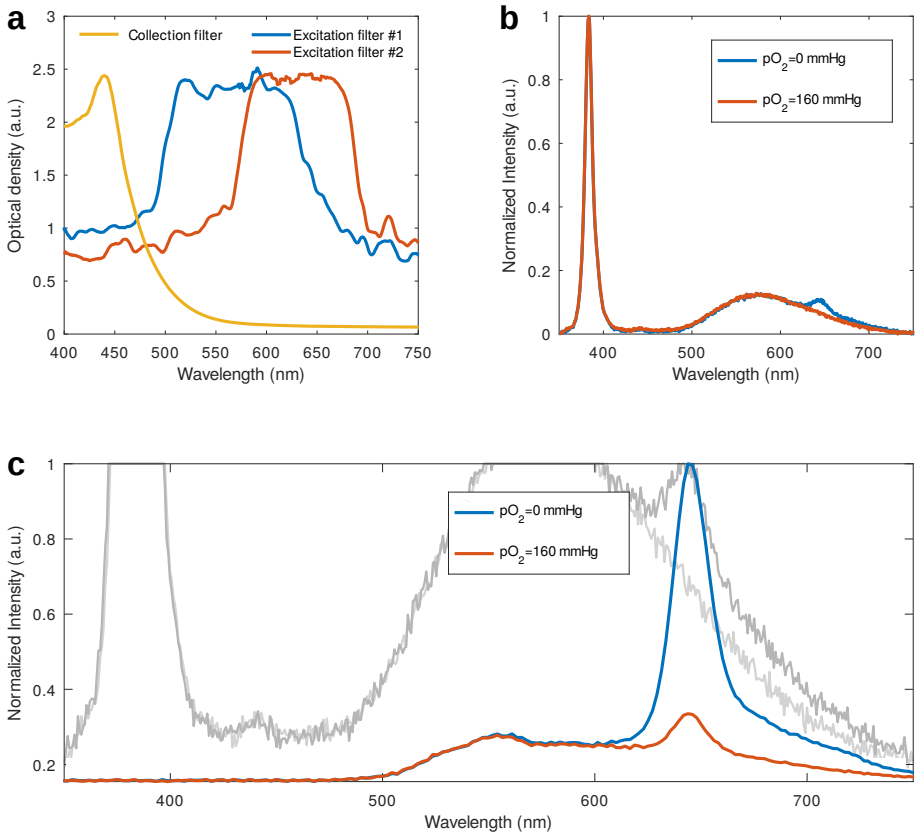

Fig. S2. (a) Optical density of the excitation and collection optical filters. (b) Optical spectrum of the pulsed LED, revealing an unwanted phosphorescent "hump", which overlaps with the emission spectra of the oxygen-sensing porphyrin. (c) The emission filters successfully filter out the unwanted light from the LEDs.

CALIBRATION SET-UP

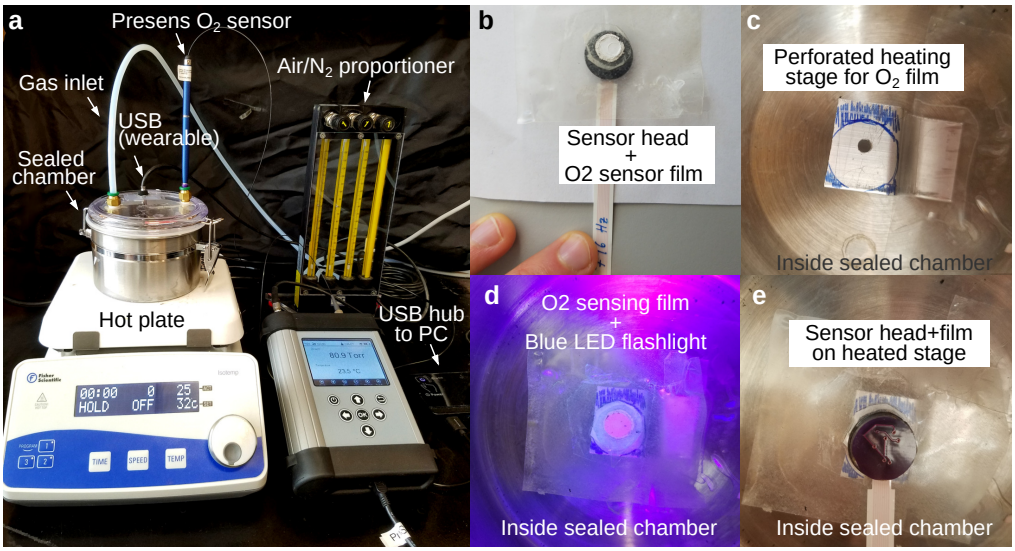

Fig. S3. Calibration set-up and mounting of the sensor head on a heating stage inside a sealed chamber.

CIRCUIT DESIGN AND CHARACTERIZATION

Schematics

Figure S4 presents the schematics of the reference signal conditioning circuit (a)(and node signals measured by an oscilloscope in (b)), the ADC connections (b), and the ribbon cable connections between the control electronics PCB (d) and the sensor head (e).

Concerning the signal conditioning circuit, the capacitor for the 4-pole low pass filter was fixed at 0.1 μF, and the resistor was chosen so that the cutoff frequency of the LP filter was centered at the fundamental frequency of the PWM signal. Therefore, $f_C = \frac{1}{2\pi RC} \Rightarrow R = \frac{1}{2\pi C \cdot f_C}$, which yields a $5k\Omega$ for a modulation frequency of 796 Hz.

The ground and V_{IN} were provided by the microcontroller, which was either the Particle Photon (USB and Wi-Fi) or the Particle Argon (USB, Wi-Fi, BLE and enough RAM to perform on-board signal analysis). The power to the microcontroller was provided by USB or by a Li-Po battery which was connected to the appropriate power pin on the microcontroller board.

Transimpedance amplifier circuit

The amplification range of the TIA application note (ref [50]) was fixed at 1 MHZ. The gain resistor was chosen to maximize the use ADC range in full so that the maximum phosphorescence signal (at 0 mmHg) would result in high bit counts but which would not saturate the ADC channel. With this resistor fixed at 2.5MΩ, and a desired amplification range of 1 MHz, the capacitor required was obtained from $C \leq \frac{1}{2\pi R_1 f_P}$ which leads to $C \leq 0.06pF$. We were able to find $C = 0.1pF$ which means that the bandwidth is 636.6kHz which is three of orders of magnitude higher than the modulation frequency (796 Hz). The DC and AC transfer function of the TIA is shown further below, showing the gain and bandwidth of the circuit.

TIA characterization

The following measurements were carried out on a prototyping board where the circuit was first built before designing the device PCBs. To characterize the DC transfer function, we moved a constant source of illumination at increasingly shorter distance on the photodiode. The current generated by the photodiode was measured by placing an ammeter in series with the photodiode, and the output of the TIA was measured by a voltmeter.

Figure S5(a) plots the result of the experiment, which shows the expected linear dependence of output voltage with photocurrent. A linear fit reveals the voltage offset of the TIA is 0.130 V (100

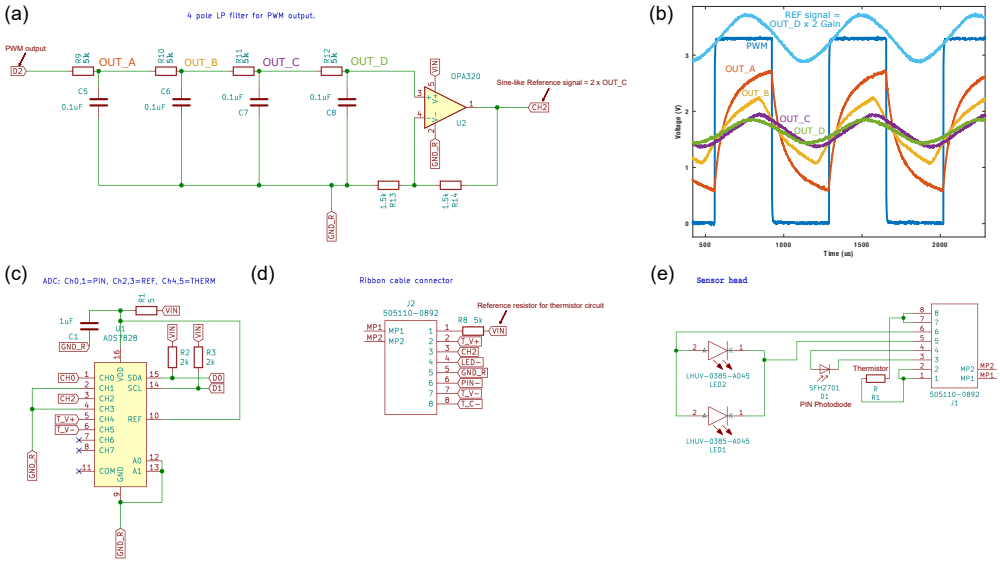

Fig. S4. Schematic of the (a) reference signal conditioning circuit. (b) Signals at different nodes of (a) measured by an oscilloscope, with a final reference signal with a mean value at 3.3 V which drives the LEDs. Schematic of (c) ADC circuit, (d) ribbon cable connections and (e) sensor head schematic.

mV per design) and the V/A gain is 2.4 $M\Omega$, which is close to the value of the R_6 resistor in the circuit schematic.

With regards to bandwidth, we followed the design on reference [50] to reproduce the 1MHz photodiode amplifier characteristics on our device. To do this, once the sensor head was built with the optimal geometry (LEDs as close as possible to the photodiode), we tweaked the gain resistor on the circuit to be able to detect the optical signal from the oxygen-sensing film at both room air (low signal) and to not saturate the amplifier at low oxygen (high signal). We settled on a $R_G = 2.5M\Omega$ gain resistor (as opposed to 53.6 $k\Omega$ in ref. [50]), which amplifies a 1.92 μA current to 4.9V. In order to keep the -3dB bandwidth at $f_P = 1$ MHz as in the reference, we needed a capacitor $C \leq 1/(2\pi R_G f_P) = 0.06pF$. We were able to buy a 0.01pF, which brings $f_P = 636.6kHz$.

To characterize the AC transfer function, we followed the schematic on page 14 of ref [50] (Texas Instrumets application note) using a Tektronix oscilloscope with built-in function generator and a photodiode simulator with a $2M\Omega$ resistor chosen to produce a $0.25\mu A$ current at the driving frequency which had peak-to-peak amplitude of 1 V. The amplitude of the output signal of the TIA was measured with the oscilloscope, and is shown in Fig.S5(b) in dB, using the equation $dB = 20log_{10}(V_{OUT}/I_{IN})$, where $I_{IN} = 0.25\mu A$. As can be seen, a steep decline of the transfer function magnitude occurs for frequencies higher than $f_P = 636.6kHz$ (marked by a red dot).

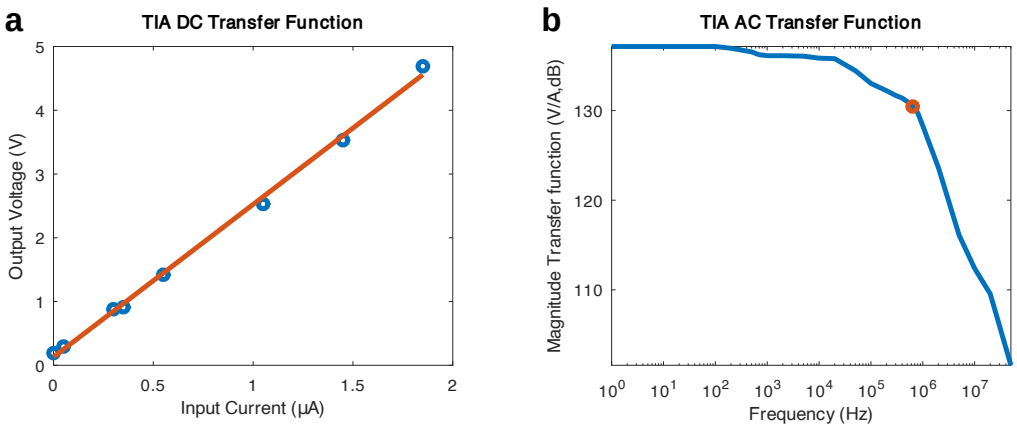

Fig. S5. DC and AC transfer function of the transimpedance amplifier circuit.

ADC Inputs and Outputs

The input signals to the ADC can be seen in Fig. S6 (top), measured by an oscilloscope, where the reference signal is shown in yellow and the photodiode signal in blue.

With regards to the output, if the analysis isn't done on-board of the device, the following JSON string can be read through serial port (done in Python in our case):

```
{"temp":[28.35], "time":[1,282,564,845,1127,1412,1694,1975,2256,2544,2826,3107,
...,281619]
"voltage":[821,61,1061,2284,1607,118,379,2005,2200,634,71,1247,2307,1435,86,
...,979]
"tclk":[142,423,705,986,1267,1553,1834,2116,2397,2685,2966,3247,3529,3810,
...,281759]
"clk":[3072,3192,3756,3843,3321,3034,3492,3873,3579,3040,3245,3815,3803,3258,
...,3]}
```

The JSON string is parsed to extract the value of temperature, and the arrays of t_{REF}, V_{REF}^{ADC} and t_{PIN}, V_{PIN}^{ADC}.

```
parsed_json = json.loads(line)
temp = parsed_json['temp'];
t = parsed_json['time'];
tclk = parsed_json['tclk'];
pin = parsed_json['voltage'];
clk = parsed_json['clk'];
```

The signals shown as bits in the Y scale are plotted in Fig. S6 (bottom left and right). A text file with the full 2000 values per array can be downloaded here.

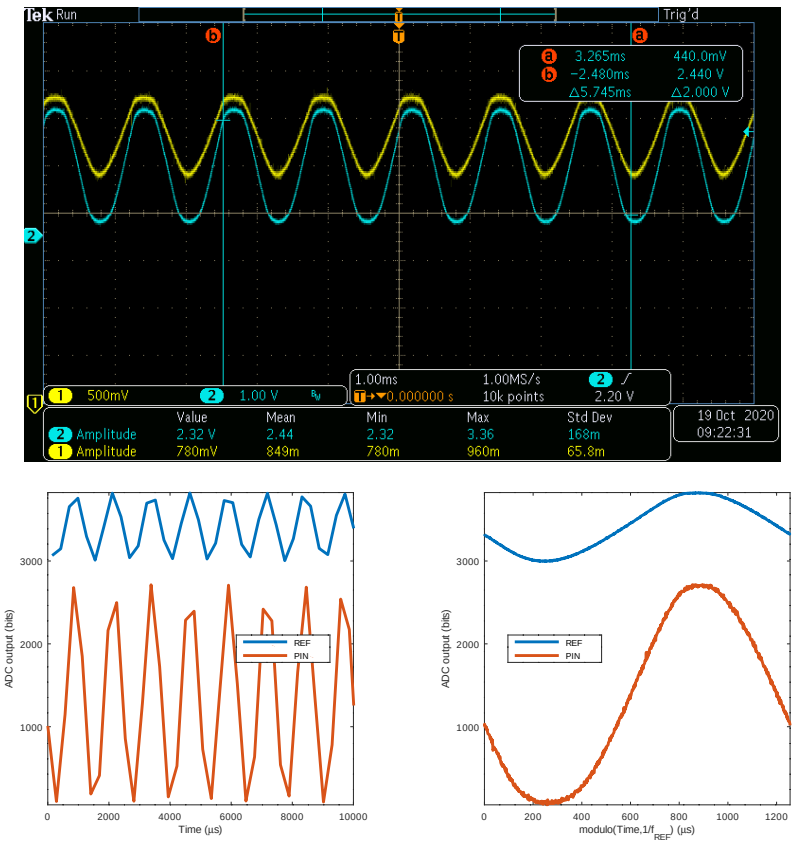

Fig. S6. Input signals to the ADC (top), reference (yellow) and photodiode (blue) measured by an oscilloscope. Output bits of ADC for the reference and photodiode channels, before (bottom left) and after modulo operation (bottom right).

Power consumption

To measure the power consumption, we modified a USB power cable to interrupt the current path and include an ammeter to measure the current. If the device is powered by USB (constant voltage source of $V = 5.17V$), we measure a current of $I_{OFF} = 38mA$ with LEDs OFF, and $I_{ON} = 108mA$ with LEDs ON, so $P_{OFF} = V \cdot I_{OFF} = 196\mu W$ and $P_{ON} = V \cdot I_{ON} = 558\mu W$. If a measurement is taken every $t_S = 5s$ and the LEDs are flashed for $t_{LED} = 0.2s$ every measurement, then the average power is

$$\overline{P} = \frac{(t_S - t_{LED}) \cdot P_{OFF} + t_{LED} \cdot P_{ON}}{t_s} = 211mW$$

Our device has not been optimized for power consumption, as the circuit scheme was designed for proof of principle, but its energy usage compares favorably to commercial wearable or portable devices, such as the Garmin Edge 800 GPS bike computer ($1200mAh$ and 15h battery life, $P = 296mW$) or the Polar V800 triathlon watch ($350mAh$ and 13h battery life, $P = 100mW$).

ADC SIGNAL PROCESSING AND FITTING

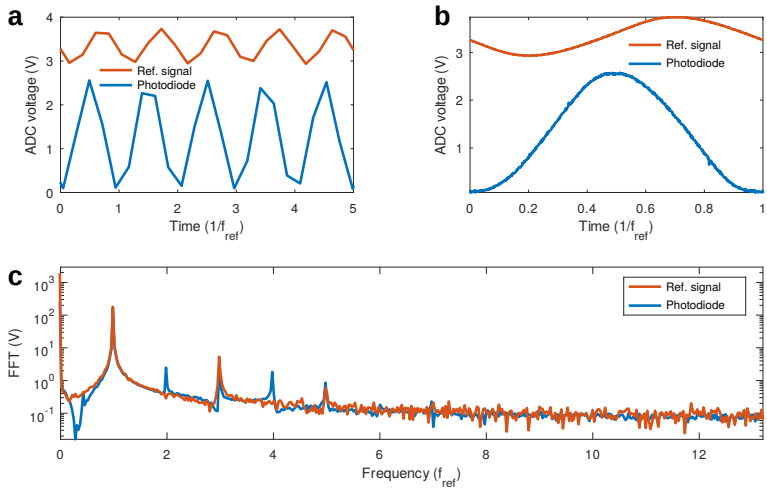

Fig. S7. (a) Photodiode and reference signal, with $f_r = 796Hz$, sampled at $\sim 5kHz$. (b) Photodiode and reference signal are sampled for 1000 points each, with time being plotted as the modulo with base $1/f_r$ (the period of the reference signal). Because our signal does not change in this timescale (0.2 s), we are able to reconstruct a single oscillation of our signals with very high detail. (c) Fast Fourier transform of the reference and photodiode signals. The reference signal reveals the presence of odd harmonics, which leak from the PWM output through the low pass filter. The photodiode signal also contains even harmonics, which could originate from the non-linear nature of the LEDs.

With regards to signal processing, the conditioning of the reference signal is described in the manuscript (4-pole low pass filter and $x2$ gain amplifying stage) and the circuit schematics shown above. The photodiode signal is directly fed into the TIA input and is not filtered through hardware.

The signals plotted in Fig. 2 correspond to the ADC output channels for the photodiode and reference signal, and this numerical data does not undergo any filtering via software. The algorithms for extracting lifetime and intensity (linear regression in the matrix form) and for predicting pO_2 from lifetime and intensity are described in full in the manuscript, and this simplicity is where the beauty (or promise) of the approach lies.

The Octave or C++ code used to perform the signal analysis uses no other ingredients. The raw waveforms, the reconstruction of a single period using the modulo operator and the harmonic content of the signals is shown in Fig.S7. A result of the fitting is shown in Fig.S8.

A snippet of the Octave code to extract phase and amplitude from the photodiode signal is shown here:

```
t=mod(t,T); % time modulo period T = 1/f_ref;
[t,I] = sort(t); %sort time ascending
x=x(I); % sort photodiode voltage wrt time

% multiple linear regression in matrix form
y=x;
ones=zeros(1,length(x))+1;
w=2*pi*ref_freq;
x1=cos(w*t-0.0*pi);
x2=sin(w*t-0.0*pi);
x3=cos(3*w*t);
x4=sin(3*w*t);
x5=cos(5*w*t);
x6=sin(5*w*t);
x7=cos(2*w*t);
```

```
x8=sin(2*w*t);

M=[ones;x1;x2;x3;x4;x5;x6;x7;x8]';

p = inv(M'*M)*M'*y'; % fitting coefficients
b0=[sqrt(p(2)^2+p(3)^2),sqrt(p(4)^2+p(5)^2),sqrt(p(6)^2+p(7)^2),sqrt(p(8)^2+p(9)^2)];
%amplitude of each harmonic
phase=[acos(p(2)/b0(1)),acos(p(4)/b0(2)),acos(p(6)/b0(3)),acos(p(8)/b0(4))]; %phase
of each harmonic
%fitting function
fit=p(1)+p(2)*x1+p(3)*x2+p(4)*x3+p(5)*x4+p(6)*x5+p(7)*x6+p(8)*x7+p(9)*x8;
```

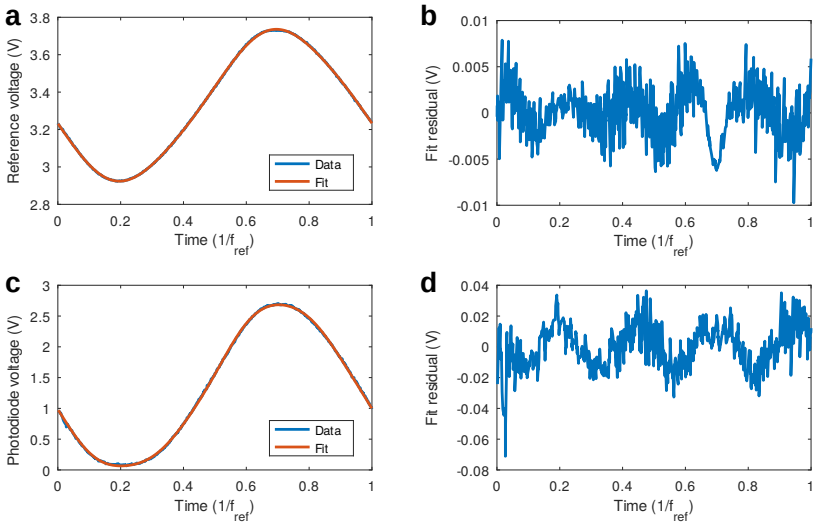

Fig. S8. Fitting of photodiode and reference with a linear combination of sines. The reference frequency is fit with the fundamental frequency and harmonics $3f$, $5f$ and $7f$, with the photodiode requiring an additional $2f$ harmonic for a proper fit.

TEMPERATURE DEPENDENCE OF THE LED LEAKAGE

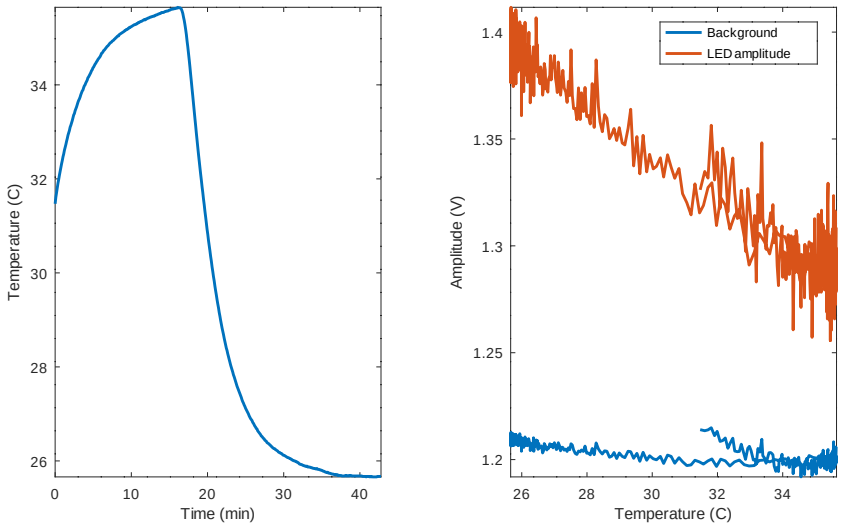

Fig. S9. Temperature dependence of the LED leakage which is measured with the photodiode using a blank control film. (a) Temperature values reached by the sensor head during the measurement. (b) Percentage change of the background (β_0 in the linear regression) and intensity of the fundamental frequency. The value for the background matches the values obtained for I_{OFF} in the main manuscript, while the LED leakage which is observed in the fundamental mode explains the difference in lifetime measured by our device with the spectrophotometer measurements. Including a linear dependence of the LED in the Stern-Volmer fit did not improve the quality of the fit and the coefficients obtained were of negligible value.

THERMISTOR CIRCUIT

A thermistor's B value, or beta value, is an indication of the shape of the curve representing the relationship between resistance and temperature of a negative temperature coefficient (NTC) thermistor.

A thermistor is defined by constant B between two temperatures, specified by the manufacturer, given by the Steinhart-Hart equation:

$$B_{T_1/T_2} = \frac{T_2 T_1}{T_2 - T_1} ln\left(\frac{R(T_1)}{R(T_2)}\right)$$

In our case, $B = 3650$, $T_1 = 25C + 273.15 = 298.15K$ and $R(T_1) = 10k\Omega$, so by measuring the resistance we can know the temperature by measuring the resistance at T_2:

$$T_2 = \frac{BT_1}{B - T_1 ln\left(\frac{R(T_1)}{R(T_2)}\right)}$$

We measure the resistance of the thermistor by powering a circuit with two resistors, (reference resistor R_1 and thermistor R_2) in series with a constant voltage and measuring the voltage drop on the thermistor, as shown in Fig. S10. Because the voltage is known and one resistor is fixed, using Ohm's law the resistance of the thermistor can be calculated knowing $T_1 = 25.0 + 273.15$, $R_1 = 5k\Omega$ reference resistor, $R_2^{25^\circ C} = 10k\Omega$; $V = 5V$, so measuring V_2 on the thermistor, we can substitute $R_2(T_2) = \frac{V_2}{V - V_2}$ in the equation above for T_2 to calculate the temperature.

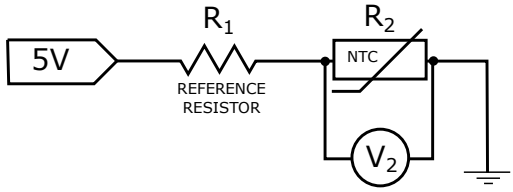

Fig. S10. Thermistor circuit.

BLE COMMUNICATION

As is mentioned in the manuscript, the multiple linear regression algorithm is implemented in the microcontroller. This allows to calculate the phase and amplitude of the signal on-board the device. The code uses the *MatrixMath.h* Arduino library to perform the same calculations described above for Octave.

Then, every time the pO_2 is sampled, a custom BLE GATT protocol is used to transmit to a PC three quantities: {TEMPERATURE, PHASE, AMPLITUDE}. This is done through a Python script which uses the *Bleak* library, and plugging in the device's Bluetooth MAC address and the UUIDs for the custom BLE services. From these three variables, and having previously performed a calibration of the oxygen-sensing film under use, the pO_2 in mmHg is calculated using the temperature-dependent Stern-Volmer equation.

The technique can be refined further by implementing the Stern-Volmer equation on the device's firmware and have the device output directly temperature and pO_2 values, but we wished to save the values of phase and amplitude during our experiments. However on most applications we chose to measure via USB as we could store the "raw" waveforms.